**有作用域的 CSS**

当 <style> 标签有 scoped 属性时，它的 CSS 只作用于当前组件中的元素。这类似于 Shadow DOM 中的样式封装。它有一些注意事项，但不需要任何 polyfill。它通过使用 PostCSS 来实现以下转换：

<style scoped>

.example {

color: red;

}

</style>

<template>

<div class="example">hi</div>

</template>

转换结果：

<style>

.example[data-v-f3f3eg9] {

color: red;

}

</style>

<template>

<div class="example" data-v-f3f3eg9>hi</div>

</template>

**注意**

**混用本地和全局样式**

你可以在一个组件中同时使用有作用域和无作用域的样式：

<style>

/\* 全局样式 \*/

</style>

<style scoped>

/\* 本地样式 \*/

</style>

**子组件的根元素**

使用 scoped 后，父组件的样式将不会渗透到子组件中。不过一个子组件的根节点会同时受其父组件有作用域的 CSS 和子组件有作用域的 CSS 的影响。这样设计是为了让父组件可以从布局的角度出发，调整其子组件根元素的样式。

**深度作用选择器**

如果你希望 scoped 样式中的一个选择器能够作用得“更深”，例如影响子组件，你可以使用 >>> 操作符：

<style scoped>

.a >>> .b { /\* ... \*/ }

</style>

上述代码将会编译成：

.a[data-v-f3f3eg9] .b { /\* ... \*/ }

有些像 Sass 之类的预处理器无法正确解析 >>>。这种情况下你可以使用 /deep/ 操作符取而代之——这是一个 >>> 的别名，同样可以正常工作。

**动态生成的内容**

通过 v-html 创建的 DOM 内容不受作用域内的样式影响，但是你仍然可以通过深度作用选择器来为他们设置样式。

**还有一些要留意**

* **CSS 作用域不能代替 class**。考虑到浏览器渲染各种 CSS 选择器的方式，当 p { color: red } 设置了作用域时 (即与特性选择器组合使用时) 会慢很多倍。如果你使用 class 或者 id 取而代之，比如 .example { color: red }，性能影响就会消除。你可以在[**这块试验田**](https://stevesouders.com/efws/css-selectors/csscreate.php)中测试它们的不同。
* **在递归组件中小心使用后代选择器!** 对选择器 .a .b 中的 CSS 规则来说，如果匹配 .a 的元素包含一个递归子组件，则所有的子组件中的 .b 都将被这个规则匹配。